IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 23, 2019, accepted March 17, 2019, date of publication April 22, 2019, date of current version May 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2912182

Overlapping Community Detection Algorithm Based on Coarsening and Local Overlapping Modularity

ZHANGHUI LIU[1,2,3], BINGJIE XIANG[1,2,3], WENZHONG GUO[1,2,3], YUZHONG CHEN[1,2,3], KUN GUO[1,2,3], AND JIANNING ZHENG[4]

[1]College of Mathematics and Computer Sciences, Fuzhou University, Fuzhou 350116, China
[2]Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350116, China
[3]Key Laboratory of Spatial Data Mining Information Sharing, Ministry of Education, Fuzhou 350116, China
[4]State Grid Info-Telecom Great Power Science and Technology Co., Ltd., Xiamen 350200, China

Corresponding author: Kun Guo (gukn123@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61300104, Grant 61300103, and Grant 61672158, in part by the Fujian Province High School Science Fund for Distinguished Young Scholars under Grant JA12016, in part by the Program for New Century Excellent Talents in Fujian Province University under Grant JA13021, in part by the Fujian Natural Science Funds for Distinguished Young Scholar under Grant 2014J06017 and Grant 2015J06014, in part by the Major Production and Research Project of Fujian Scientific and Technical Department, Technology Innovation Platform Project of Fujian Province, under Grant 2009J1007 and Grant 2014H2005, in part by the Fujian Collaborative Innovation Center for Big Data Applications in Governments, in part by the Natural Science Foundation of Fujian Province under Grant 2013J01230 and Grant 2014J01232, in part by the Fujian Industry-Academy Cooperation Project under Grant 2014H6014 and Grant 2017H6008, and in part by the Haixi Government Big Data Application Cooperative Innovation Center.

ABSTRACT Community detection is an important research direction in the field of complex network analysis. It aims to discover community structures in complex networks. Algorithms based on dynamic distance mechanism can find stable communities with various shapes. However, they still cannot discover overlapping or outlier communities. This paper proposes an overlapping community discovery algorithm based on coarsening and local overlapping modularity. First, to reduce the running time, a new equation for computing the local overlapping modularity increment is derived. This equation finds the overlapping communities, accurately and quickly. Second, a new similarity measuring strategy is designed to reduce the number of outlier communities. The experiments on artificial and real datasets show that the proposed algorithm can discover the overlapping communities, accurately and efficiently.

INDEX TERMS Community detection, overlapping communities, triangle coarsening, local overlapping modularity.

I. INTRODUCTION

In the real world, the relationships among many things are complicated and can be represented by complex networks. Therefore, complex network analysis has become a trending research topic. Complex networks, such as literature reference networks, scientific collaboration networks, protein–protein interaction networks, and urban transportation networks, have become the research subjects of many scholars. As an important feature of complex networks, community structure is characterized by the tight connections among nodes inside the same communities and the loose connections among nodes in different communities [1]. The goal of community discovery is to discover communities from complex networks efficiently and accurately. With community discovery, researchers can understand the internal mechanisms of complex networks well. This understanding has great value to our everyday life.

Early community detection algorithms can only discover non-overlapping communities. However, in the real world, a node can belong to multiple communities. For example, a person can both be a member of his/her friendship and colleague circles. Hence, detecting overlapping communities is important to reveal complex community structures. At present, overlapping community detection algorithms can be divided various types, such as label propagation-based

The associate editor coordinating the review of this manuscript and approving it for publication was Zhan Bu.

algorithms (LPAs), seed expansion-based algorithms, and edge clustering-based algorithms. LPAs have the merits of low computation complexity and easy parallelization, though they can easily create fragment communities. Generated communities are also unstable. Seed expansion-based algorithms are capable of generating communities with various shapes. However, they are sensitive to the selection of seeds. Edge clustering-based algorithms can discover overlapping naturally but have high time complexity. Recently, a series of community detection algorithms based on dynamic distance was proposed. These algorithms achieve community detection by computing the influence of neighbors to nodes' distance to determine whether edges are preserved or cut off. The type of algorithms is based on a simple but effective theory that can find stable communities. However, their convergence speed is slow. To solve the problem, an overlapping community detection algorithm based on coarsening and local overlapping modularity (CDCLM) is proposed. It can quickly detect overlapping communities by introducing the coarsening strategy and the local overlapping modularity.

The main contributions of this paper are as follows.

(1) the equation of the local incremental overlapping modularity is deduced to discover the overlapping communities accurately and quickly. Existing algorithms based on traditional modularity either fail to find overlapping communities or require global information. The proposed equation improves the efficiency of modularity calculation and the quality of the overlapping communities.

(2) A new community optimization strategy is adopted to merge similar communities and reduce the number of outlier communities. Existing algorithms without the optimization may result in excessive community overlaps. The new strategy can improve the accuracy of community division.

(3) The comprehensive experiments conducted on the artificial and real datasets demonstrate the effectiveness and practicability of the proposed algorithm.

II. RELATED WORK

Algorithms for overlapping community detection can be divided into different types, including methods based on label propagation [2]–[7], methods based on generative model [8]–[11], link-based methods [12]–[16], methods based on seed expansion [17]–[23], methods based on game theory [24]–[26], methods based on dynamic distance [27]–[30], and so on.

The basic idea of label propagation algorithms is to initialize a unique label for each node and update their labels iteratively. Nodes with the same label are in the same community. In 2007, Raghavan et al. first proposed the RAK algorithm for community detection. The primary idea is to assign a node to the label adopted by most of its neighbors. When each node's label is iteratively updated until it no longer changes, nodes with the same label are grouped into the same community [2]. Gregory proposed the COPRA algorithm to make LPA applicable to overlapping community detection [3]. Xie et al. proposed a label propagation algorithm based on neighborhood strength driving. The algorithm adopts a new update strategy to improve the computation efficiency and community partition quality [4]. In 2016, Zhang et al. proposed the COPRA-EP algorithm to improve the COPRA algorithm by solving the problem of unstable community detection results [5].

Algorithms based on a generative model assume that the community relationships among nodes obey certain distributions. Statistical inference methods are used to find the optimal parameters of the distributions and obtain the communities' generation models. Newman et al. proposed a hybrid probability model to describe the community structures and used EM algorithms to find overlapping communities [8]. Airoldi et al. proposed a model of mixed membership stochastic block (MMSB). The model extends block models for relational data to ones that capture mixed membership latent relational structure. Thus, it provides an object-specific low-dimensional representation [9]. Xing et al. proposed a dynamic model of the MMSB (dMMSB) that can analyze the dynamic tomography of time-evolving networks [10]. In 2016, Xin et al. proposed the ARWS algorithm, which can adaptively update the affected nodes and communities when dynamic events occur [11].

Algorithms based on link clustering aim to transform the nodes of the networks into edges and convert discovered edge clusters into node clusters. In 2004, Pereira-Leal et al. first proposed a link clustering algorithm that applied to group identification in protein networks [12]. In 2013, Shi et al. proposed the GaoCD algorithm that applied genetic operation to link clustering [13]. Zhu et al. proposed a new density-based link clustering algorithm called DBLINK. Their proposed algorithm can identify isolated edges and assign them to relevant communities to improve the accuracy of community discovery [14]. He et al. proposed a generative model for link clustering and the NMFIB algorithm. The algorithm depends on the importance of each node to describe the structure of link communities when forming links in each community [15]. Guo et al. proposed the OCDEDC algorithm, which solves the problems of obscure belongingness of the nodes on community borders and the excessive overlap of communities [16].

Algorithms based on seed expansion depend on the design of certain strategies to find seed nodes or seed communities in the networks. The seeds are then expanded in accordance with the local information to discover communities. In 2008, Lancichinetti et al. proposed the LFM algorithm based on local optimization [18]. Kanawati et al. proposed the Licod algorithm based on the concept of community member preference list [19]. In 2017, Su et al. proposed the RWA algorithm that used the strategy of random walk for seed expansion [22].

In 2010, Chen et al. applied the game theory to community detection [24]. Algorithms based on game theory model the community formation process as a community formation game. Each node increases its profit by joining and leaving and transforming communities until the algorithm reaches the equilibrium status and discovers the final communities. In 2017, Bu et al. proposed the SLA algorithm.

TABLE 1. Overview of the variables.

Variables	Details
$NB(u)$	neighbors of node u without itself
$\Gamma(u)$	neighbors of node u including itself
$NB(C_i)$	neighbors of community C_i
$d(u,v)$	jaccard distance between node u and node v
$CN(u,v)$	common neighbors of nodes u and v
$EN(u,v)$	The exclusive neighbors between nodes u and v
$intimacy(C_l,C_k)$	intimacy of communities C_l and C_k
$\rho(x,u)$	influence factor of the distance of nodes u and v
DI	influence of direct neighbors
CI	influence of common neighbors
EI	The influence of exclusive neighbors

The algorithm can start from arbitrary initial clusters and find the corresponding balanced solution of attributed graph clustering, where all nodes and clusters are satisfied with the final cluster configuration [25]. In 2019, Bu et al. proposed a novel and powerful graph K-means framework. It can effectively integrate both the topological and the attributive information in SMNs for community detection [26].

Algorithms based on dynamic distance automatically spots communities in the networks by verifying the changes of the ''distance'' between nodes (i.e. dynamic distance). The idea of the algorithms is to envision the target network as an adaptive dynamical system where each node interacts with its neighbors. The interaction changes the distance between nodes and the changed distance affects the interactions. The interplay eventually leads to a stable distribution of node distance. Nodes in the same community move towards each other. Nodes in different communities keep far away from each other. The nodes with 0 distance are assigned to the same communities. In 2015, Shao et al. proposed the Attractor algorithm based on dynamic distance [27]. In 2016, Meng et al. proposed the I-Attractor algorithm. This algorithm solves the problem of the Attractor algorithm, which ignores the cohesiveness difference of each node's neighbors and converges slowly [28]. In 2018, Chen et al. proposed the L-Attractor algorithm. The Attractor algorithm cannot detect the overlapping structure. L-Attractor tries to solve the problem by transforming the original graph to a link graph [29]. In 2019, Xiang et al. proposed the CDTD algorithm. The algorithm uses the triangle roughening strategy and the similarity strategy to enhance its time efficiency [30].

III. PRELIMINARIES

Given an undirected graph $G = (V, E, W)$ where V denotes the set of nodes, E denotes the set of edges, W denotes the set of weight. An overview of the variables is shown in Table 1.

Definition 1: $NB(C_i)$ is defined as

$$NB(C_i) = \cup_{v \in C_i} NB(v) - C_i. \tag{1}$$

Definition 2: The Jaccard distance between node u and node v, $d(u, v)$, is defined as

$$d(u, v) = 1 - \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}, \tag{2}$$

where $|*|$ indicates the size of set $*$.

For a weighted undirected graph, the Jaccard distance between node u and node v, $d(u, v)$, is defined as

$$d(u, v) = 1 - \frac{\sum\limits_{x \in (\Gamma(u) \cap \Gamma(v))} (w(u, x) + w(v, x))}{\sum\limits_{(x,y) \in E; x,y \in (\Gamma(u) \cup \Gamma(v))} w(x, y)}. \tag{3}$$

Definition 3: Suppose an edge exists between node u and node v, the common neighbors of the nodes, $CN(u, v)$, are

$$CN(u, v) = NB(u) \cap NB(v). \tag{4}$$

Definition 4: Suppose an edge exists between node u and node v, the exclusive neighbors of node u, $EN(u)$, are

$$EN(u) = NB(u) - CN(u, v). \tag{5}$$

Definition 5: Suppose the initial community division is $C = \{C1, C2, \ldots, Cm\}$, the similarity between community C_l and community C_k is defined as

$$intimacy\,(C_l, C_k) = \frac{|NB(C_l) \cap NB(C_k)|}{min(|NB(C_l)|\,, |NB(C_k)|)}. \tag{6}$$

Definition 6: Suppose an edge exists between node u and node v as well as node $x \in NB(u)$, the influence factor of node x on distance $d(u, v)$ is defined as

$$\rho(x, u) = \begin{cases} 1 - d(x, v) & if\ (1 - d(x, v) \geq \lambda) \\ 1 - d(x, v) - \lambda & otherwise. \end{cases} \tag{7}$$

where λ is called the cohesion factor. This factor is used as the threshold for judging the influence factor of exclusive neighbors.

Definition 7: Suppose an edge exists between node u and node v, the influence of node u and node v on distance $d(u, v)$ is defined as

$$DI = -(\frac{f(1 - d(u, v))}{ku} + \frac{f(1 - d(u, v))}{kv}). \tag{8}$$

Definition 8: Suppose an edge exists between node u and node v, the influence of the common neighbors of node u and node v on the distance $d(u, v)$ is defined as

$$CI = -\sum_{x \in CN(u,v)} \left(\begin{array}{l} \frac{f(1 - d(x, u)) \cdot (1 - d(x, v))}{ku} \\ + \frac{f(1 - d(x, v)) \cdot (1 - d(x, u))}{kv} \end{array} \right). \tag{9}$$

Definition 9: Suppose an edge exists between node u and node v, the influence of the exclusive neighbors of node u and node v on distance $d(u, v)$ is defined as

$$EI = -\sum_{x \in EN(u)} \frac{f(1 - d(x, u)) \cdot \rho(x, u)}{ku} - \sum_{y \in EN(v)} \frac{f(1 - d(y, v)) \cdot \rho(y, v)}{kv}. \tag{10}$$

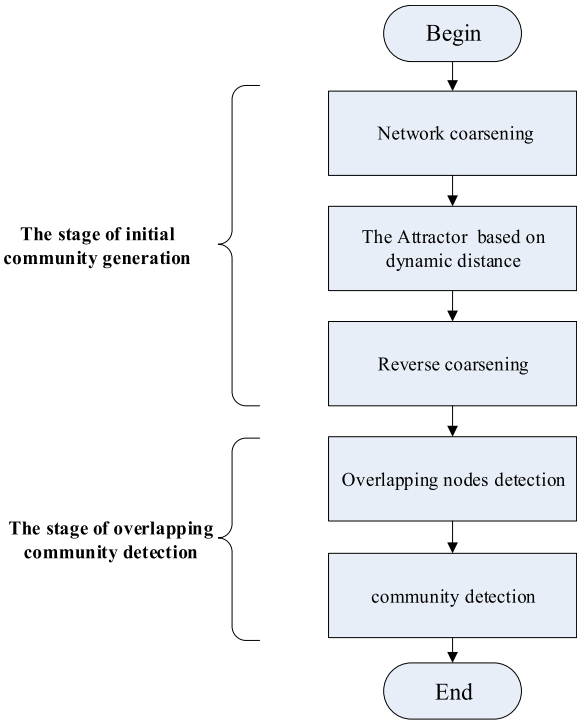


FIGURE 1. Schematic of the CDCLM algorithm.

IV. COMMUNITY DETECTION BASED ON COARSENING AND LOCAL MODULARITY

A. FRAMEWORK OF THE ALGORITHM

As shown in Fig. 1, the CDCLM runs in two stages:

Stage 1 (Initial Community Generation): The stage mainly contains three steps.

Step 1 - network coarsening. The network is coarsened according to the triangle structure characteristics. The endpoints of each traversed triangle are combined into one node. A triangle is a three-complete graph that can be regarded as the smallest stable community; as such, the coarsened triangle structures exhibit strong community characteristics [31]. In this manner, the community feature of the compound node is consistent with that of the combined nodes. As an important feature of complex networks, community structure requires that the nodes in a community be closely connected and that the edge density of the community be high. Generally, a triangle's endpoints always belong to a community or the overlapping areas among communities. They are less likely to belong to different communities.

Step 2 - running Attractor. The Attractor algorithm with dynamic distance is used to discover the initial communities.

Step 3 - reverse coarsening. The communities generated by Step 2 are restored to the communities in the original network. The nodes that constitute each compound node are assigned to the community of the corresponding compound node to get the final communities.

Stage 2 (Overlapping Community Detection): First, the community belongingness of the boundary nodes is determined based on the incremental overlapping modularity. Nodes that belongs to multiple communities are called overlapping nodes. Second, the closeness of the outlier communities to other communities is calculated to further improve the community quality. The outlier communities are merged with the closest communities.

The framework of the CDCLM algorithm is summarized as follows.

Algorithm 1 CDCLM

Input: network $G = (V, E, W)$, threshold δ
Output: community set C'
1: $C = originalComDetection(G, \delta)$;
// The stage of initial community division
2: $C' = overlapComDetection(G, C)$;
// The stage of overlapping community detection
3: OUTPUT C'

In Algorithm 1, threshold δ represents the coarsening rate in the first stage. Function $originalComDetection(G, \delta)$ is used to discover communities based on triangle coarsening and the Attractor algorithm with dynamic distance. Function $overlapComDetection(G, C)$ is responsible for discovering overlapping communities by local incremental modularity calculation. The details of the functions are described in the following sections.

B. INITIAL COMMUNITY GENERATION

The section describes the three steps of initial community generation processing in detail.

(1) First, the initial network G is traversed. For each triangle encountered in the network, the endpoints of the triangle are fused into a composite node. Second, the neighbors of the composite node and the weights of its edges are updated, and the edge relationship mapping of the endpoints is fused into the composite node. Specifically, the coarsening graph G_1 is generated after the first layer is coarsened. Second, G_1 is used as the initial network for the second layer coarsening. The above steps are repeated to generate the subsequent coarsened networks. When the coarsening rate $(G_m - G_{m-1}) / G$ of the m-th coarsening is smaller than the given coarsening rate threshold δ, the coarsening ends. The final coarsened graph G' is the output.

(2) The Attractor mechanism with dynamic distance is used to discover the community structure of the coarsening network. First, the distances of the endpoints of all the edges are initialized in accordance with Equation (2). Then, they are updated based on three types of neighbor influence. Neighbor influence can be divided into three types.

(a) *Influence of direct neighbors, DI.* This type reflects the influence of node u and node v on distance $d(u, v)$, which is calculated according to Equation (8).

(b) *Influence of common neighbors, CI.* This type reflects the influence of the common neighbors of node u and node v on distance $d(u, v)$, which is calculated according to Equation (9).

(c) *Influence of exclusive neighbors, EI*. This type reflects the influence of the exclusive neighbors of node u and node v on distance $d(u, v)$, which is calculated according to Equation (10).

The distance will converge finally to 0 or 1. The edges with the distance of 1 are cut off. The nodes in the same component are grouped in the same community.

(3) The reverse operation is used to restore the final community structures. First, the communities of composite nodes are restored to the communities of nodes in the original network according to the map containing the relations between the coarsened nodes and the original nodes generated by Step 1. Second, the initial community set C of the original network is obtained by combining the relation with the initial community set in the coarsened network returned in Step 2.

The details of initial community generation are shown as follows.

The input of function 1, network $G(V, E, W)$, is an undirected weighted graph. If an unweighted network exists, then the weight of each edge is 1. The function $Coarsening(G, \delta)$ achieves network coarsening, that is, traversing the triangle in the network, combining the three nodes of the triangle to one node, and finally obtaining the coarsening network. δ is used to control the threshold of network coarsening. The value of δ is among 0–1. The function *findComponent*(G') is used to find the component of G'. The function regress(G_{coar}) is adapted to reverse, which restores the communities of composite nodes to the communities of nodes in the original network.

C. OVERLAPPING COMMUNITY DETECTION

1) UPDATE OF INCREMENTAL OVERLAPPING MODULARITY

In 2005, Shen et al. proposed the concept of overlapping modularity (EQ), which is based on traditional modularity [1] for evaluating the quality of overlapping community detection [32]. The closer the value of EQ is to 1, the higher the quality of the communities discovered by the algorithm. The equation of EQ is as follows:

$$EQ = \frac{1}{2m}\sum_{l=1}^{c}\sum_{i,j\in C_l}\frac{1}{O_i O_j}[A_{ij} - \frac{k_i k_j}{2m}], \tag{11}$$

where m is the number of edges in the network, c is the number of communities, C_l is the lth community, O_i is the number of communities to which node i belongs, k_i is the degree of node i, and A_{ij} is used to indicate the connection between node i and node j. If an edge connecting these two nodes does exist, then A_{ij} takes a value of 1; otherwise, it is 0.

At present, many algorithms are being developed based on the global perspective of modularity optimization. Therefore, these algorithms must calculate the incremental modularity of all communities after the communities of the node changes. The measure causes the time complexity to be quite high. In fact, the local perspective considers the incremental modularity of the community associated with each node before and after the change of its membership relationship when it joins (or leaves) a community. In this manner, the calculation cost can be greatly reduced. On this basis, Equation (11) is

Function 1 originalComDetection(G, delta)

Input: network $G = (V, E, W)$, threshold δ
Output: initial community set C

```
G' = Coarsening(G, δ);          // network coarsening
FOR EACH e = {u, v} ∈ E' DO
    Using Equation (2) to calculate d(u, v);
    FOR EACH x ∈EN(u) DO
        Using Equation (2) to calculate d(u, x);
    END FOR
    FOR EACH y ∈EN(v) DO
        Using Equation (2) to calculate d(v, y);
    END FOR
END FOR
flag = TRUE;   // indicates whether there is a change in the distance during the iteration. TRUE if changed, otherwise FALSE
loop = 0;                       //Number of iteration
WHILE (flag) and (loop < 1000) DO
    flag = FALSE;
    FOR EACH e = {u, v}∈ E' DO
        Using Equation (8)(9)(10)
            to calculate DI,CI,EI, respectively;
        dist = d(u, v) + DI +CI + EI;
        IF (0 < d(u, v) < 1) THEN
            IF (dist > 1) THEN
                d(u, v) = 1;
            END IF
            IF (dist < 0) THEN
                d(u, v) = 0;
            END IF
            IF (0 < dist < 1) THEN
                d(u, v) = dist;
            END IF
            flag = TRUE;
        END IF
    END FOR
    loop = loop + 1;
END WHILE
FOR EACH e = {u, v}∈ E' DO
    IF (d(u, v) < 1) THEN
        d(u, v) = 0;
    END IF
    IF (d(u, v) >= 1) THEN
        G' = G' - e;
    END IF
END FOR
C_coar = findComponent(G');     // Discover the connected components on G' and add it as a community to C
C = regress(C_coar);            // use the reverse operation to restore the final community structure
RETURN C;
```

expressed as Equation (12) with the node's local overlapping modularity.

$$EQ = \frac{1}{2m}\sum_{i} g_i$$
$$g_i = \sum_{s}\sum_{j\in Cs}\frac{1}{OiOj}(Aij - \frac{kikj}{2m}), \tag{12}$$

where g_i is the local modularity of node i, which represents the contribution of node i to the modularity of the entire network community division from a local perspective; and C_s describes the communities of node i.

For $\forall\, i \in V$, g_i is only related to the community C_s, and the global overlapping modularity EQ increases monotonically with g_i. To prove this assumption, we submit the following proposition.

Proposition 1: Let the current community of network $G(V, E)$ be C. For $\forall i \in V$, suppose that the community set to which node i belongs changes from s_i to s_j after an iteration, and node i leaves community set $P = s_i - s_j = \{P_1, \ldots, P_n\}$ and joins community set $T = s_j - s_i = \{T_1, \ldots, T_q\}$. Then, the incremental modularity of the network is

$$\Delta EQ = \frac{\Delta g_{ii}}{m} + \frac{1}{2m}\sum_{w}\sum_{x\in P_w}\frac{O_i' - O_i}{OxO_iO_i'}(Axi - \frac{kxki}{2m}). \tag{13}$$

Proof: For any node i, if the community in which it is located changes, then the value of its corresponding g_i will also change. Therefore, the changes in the label set of i will cause the changes of the values of g_is corresponding to the nodes in $H = P_1 \cup \cdots \cup P_n \cup T_1 \cup \cdots \cup T_q$. Hence, the nodes in H can be divided into three types, and the values of g_is can be updated as follows:

(a) $\forall x \in H \&\& x \neq i, x \in P_w$ $(w \in \{1, \ldots, n\})$, the local incremental overlapping modularity Δg_{ix} can be calculated as

$$\Delta g_{ix} = -\frac{1}{OxO_i'}(Axi - \frac{kxki}{2m}). \tag{14}$$

(b) $\forall y \in H \&\& y \neq i, y \in T_r$ $(r \in \{1, \ldots, q\})$, the local incremental overlapping modularity Δg_{iy} can be calculated as

$$\Delta g_{iy} = \frac{1}{OyO_i'}(Ayi - \frac{kyki}{2m}). \tag{15}$$

(c) For node i, the local incremental overlapping modularity Δg_{ii} can be calculated as

$$\Delta g_{ii} = \sum_{r}\sum_{e\in T_r}\frac{1}{OeO_i'}(Aei - \frac{keki}{2m}) - \sum_{w}\sum_{h\in P_w}\frac{1}{O_hO_i}(Ahi - \frac{khki}{2m}) \tag{16}$$

Therefore, the derivation of change of value, Δg_i, in the entire network caused by the change of label set of the node i is as follows in Δg_i, as shown at the top of the next page.

In general, the change of EQ in the overall network caused by the change of the label set of the node i can be calculated as follows:

$$\begin{aligned}\Delta EQ &= EQ(C') - EQ(C)\\ &= \frac{1}{2m}\sum_{i}\Delta g_i\\ &= \frac{\Delta g_{ii}}{m} + \frac{1}{2m}\sum_{w}\sum_{x\in P_w}\frac{O_i' - O_i}{OxO_iO_i'}(Axi - \frac{kxki}{2m}).\end{aligned}$$

□

Therefore, unlike the traditional EQ update strategy, the CDCLM algorithm only considers the local information associated with the node when judging the influence of the network structure on node i. As a result, it reduces the time complexity of the algorithm. The original overlapping modularity is calculated by Equation (12). The global incremental overlapping modularity is obtained by calculating the incremental modularity of all communities after the community membership of node changes. The process needs to traverse all the edges in the community. Thus, the time complexity is O(m). Equation (16) is obtained by updating incremental overlapping modularity. The process only needs to consider the incremental overlapping modularity of the community where i is related. The total time complexity is O(n_i), where n_i indicates the number of nodes in the community where i is located, $n_i << m$. Therefore, the updated incremental overlapping modularity reduces the cost of calculation and greatly reduces the time complexity.

2) IMPLEMENTATION OF THE OVERLAPPING COMMUNITY DETECTION

The phase detects overlapping communities in the initial communities of the original network. First, we evaluate the community ownerships of the nodes in accordance with the incremental local overlapping modularity caused by the label change of the node. Second, we conduct community optimization, calculate the intimacy of outlier communities to other communities, identify the most intimate communities, and incorporate outlier communities into the communities.

The stage is implemented as a function as follows.

In Function 2, *findBoundaryNode*(G) is used to find the boundary node set, which is the set of the responding node with distance 1.

D. TIME COMPLEXITY

Let the number of nodes of the original network be n, the number of edges be m, the average degree of nodes be k, the number of nodes after coarsening be n', and the number of edges after coarsening be m'.

In Function 1, the time complexity of traversing triangles iteratively of step 1 is O(nlogn + Tk^2m), where T is the number of iterations of the coarsening. The time complexity of initializing the distance of nodes from step 2 to 10 is O($m' + k'm'$). The time complexity of traversing the edges

$$
\begin{aligned}
\Delta g_i &= \sum_w \sum_{x\in P_w} \Delta g_{ix} + \sum_r \sum_{y\in T_r} \Delta g_{iy} + \Delta g_{ii} \\
&= \begin{bmatrix} -\sum_w \sum_{x\in P_w} \frac{1}{OxO'_i}(Axi - \frac{kxki}{2m}) + \sum_r \sum_{y\in T_r} \frac{1}{OyO'_i}(Ayi - \frac{kyki}{2m}) \\ + \sum_r \sum_{e\in T_r} \frac{1}{OeO'_i}(Aei - \frac{keki}{2m}) - \sum_w \sum_{h\in P_w} \frac{1}{O_hO_i}(Ahi - \frac{khki}{2m}) \end{bmatrix} \\
&= \begin{bmatrix} -\sum_w \sum_{x\in P_w} \left[\frac{1}{OxO_i}(Axi - \frac{kxki}{2m}) - \begin{bmatrix} \frac{1}{OxO_i}(Axi - \frac{kxki}{2m}) \\ -\frac{1}{OxO'_i}(Axi - \frac{kxki}{2m}) \end{bmatrix} \right] \\ + \sum_r \sum_{y\in T_r} \frac{1}{OyO'_i}(Ayi - \frac{kyki}{2m}) + \sum_r \sum_{e\in T_r} \frac{1}{OeO'_i}(Aei - \frac{keki}{2m}) \\ - \sum_w \sum_{h\in P_w} \frac{1}{O_hO_i}(Ahi - \frac{khki}{2m}) \end{bmatrix} \\
&= \begin{bmatrix} -\sum_w \sum_{x\in P_w} \left[\frac{1}{OxO_i}(Axi - \frac{kxki}{2m}) - \frac{O'_i - O_i}{OsO_iO'_i}(Axi - \frac{kxki}{2m}) \right] \\ + \sum_r \sum_{y\in T_r} \frac{1}{OyO'_i}(Ayi - \frac{kyki}{2m}) + \sum_r \sum_{e\in T_r} \frac{1}{OeO'_i}(Aei - \frac{keki}{2m}) \\ - \sum_w \sum_{h\in P_w} \frac{1}{O_hO_i}(Ahi - \frac{khki}{2m}) \end{bmatrix} \\
&= \begin{bmatrix} -\sum_w \sum_{x\in P_w} \frac{1}{OxO_i}(Axi - \frac{kxki}{2m}) + \sum_r \sum_{y\in T_r} \frac{1}{OyO'_i}(Ayi - \frac{kyki}{2m}) \\ + \sum_r \sum_{e\in T_r} \frac{1}{OeO'_i}(Aei - \frac{keki}{2m}) - \sum_w \sum_{h\in P_w} \frac{1}{O_hO_i}(Ahi - \frac{khki}{2m}) \end{bmatrix} \\
&\quad + \sum_w \sum_{x\in P_w} \frac{O'_i - O_i}{OxO_iO'_i}(Axi - \frac{kxki}{2m}) \\
&= 2\Delta g_{ii} + \sum_w \sum_{x\in P_w} \frac{O'_i - O_i}{OxO_iO'_i}(Axi - \frac{kxki}{2m})
\end{aligned}
$$

of the network from step 1 to 2 is O(m'). The time complexity of traversing the exclusive neighbors of two adjacent nodes from step 4 to 10 is O($k'm'$), where k' is the average number of exclusive neighbors of two adjacent nodes. The time complexity of iteratively updating the distance of nodes from step 13 to 32 is O($T'm'$), where T' is the number of iterations. The time complexity of the loop from step 33 to 40 is O(m'). Hence, the total time complexity from step 2 to 40 is O($2m'+k'm'+T'm'$). Step 42 of reverse coarsening only needs to traverse each node and restore the composite nodes. As a result, the time complexity is O(n). Hence, the total time complexity of the function 1 is O($n\log n+Tk^2m+2m'+k'm'+T'm'+n$).

Let the number of boundary nodes be n_b, and the number of communities be n_c. In Function 2, the time complexity of traversing nodes to find boundary nodes in step 1 is O(n_b). Step 3 is used to obtain the corresponding community. On this basis, the incremental modularity reaches the maximum after node v joins. Given that the communities of the neighbors of node v is considered, the time complexity is O(k). The time complexity of step 2 to 4 is O(kn_bn_i). The time complexity of step 6 to 11 is O(n_c). Hence, the time complexity of Function 2 is O($kn_b + n_c$).

In summary, the total time complexity of the CDCLM algorithm is O($n\log n + Tk^2m + 2m' + k'm' + T'm' + n + kn_b + n_c$). For general complex networks, $m' < m$, $k' < k$, T, $T' << m$, n_b, $n_c << m'$, and $n < m$. Therefore, the time complexity of the CDCLM algorithm can be reduced to O ($n\log n + m$).

V. EXPERIMENTS

To verify the performance of the CDCLM algorithm, multiple artificial networks and real networks are used in the experiments. The hardware and software of the experiments are as follows: a PC with 3.1 GHz Pentium 4 CPU, 12 G RAM, 64 bit, and Windows 7 64bit. The codes of all algorithms are implemented in Python 3.6.

Function 2 overlapComDetection (G, C)

Input: network $G = (V, E, W)$, initial community set C
Output: final community set C'

$Node_{over} = findBoundaryNode(G)$;
FOR EACH $v \in Node_{over}$ DO // judgment of boundary nodes
  $community = \overset{com}{\text{argmax}}\, \Delta EQ$;
  // Using Equation (11) to calculate ΔEQ
  $community = community \cup \{v\}$;
END FOR
FOR EACH $c_1 \in C$ DO // merge intimate communities
  IF $c_1.size <= 5$ THEN
    $c_2 = \overset{com}{\text{argmax}}\{intimacy(c_1, com)\}$;
    // com is the element of C
    $newc_2 = c_1 \cup c_2$;
    $C' = C - c_1 - c_2 \cup \{newc_2\}$
  END IF
END FOR
RETURN C';

TABLE 2. Information of real-world datasets.

Real-world dataset	NODES	Edges	Average degree
Karate	34	78	4.59
Dolphin	61	158	5.18
Polbooks	105	441	8.40
Football	115	613	10.83
Texas	187	328	3.51
Cornell	195	301	3.09
Cora	2708	5429	4.01
Power	4941	6594	2.67
CA-GrQc	5241	14484	2.76
Astro-ph	18772	198110	21.11
CA-CondMat	23133	93439	8.08

A. EXPERIMENT DATASETS

1) REAL-WORLD DATASETS

To test the performance of the CDCLM algorithm on real-world datasets, eight real networks are selected for comparison, namely, Karate, a Zachary karate club network [33]; Polbooks, a network based on pages of books on American politics sold on Amazon [1]; Dolphin, an undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound [34]; Football, a network of American football games between Division IA colleges during regular season Fall 2000 [35]; Texas [36] and Cornell [36], the WebKB dataset consisting of 877 scientific publications classified into one of five classes; Cora, a dataset consisting of 2708 scientific publications classified into one of seven classes [37]; Power, an undirected, unweighted network representing the topology of the Western States Power Grid of the United States [38]; and CA-GrQC, the collaboration network of Arxiv General Relativity [39]; Astro-ph, Collaboration network of Arxiv Astro Physics [39]; CA-CondMat, Collaboration network of Arxiv Condensed Matter category [39]. The information of the networks is shown in Table 2.

TABLE 3. Parameter description of the artificial networks.

Parameter	Description
N	Number of nodes
k	Average degree of nodes
k_{max}	Maximum degree of nodes
C_{min}	Minimum number of nodes of community
C_{max}	Maximum number of nodes of community
on	Number of overlapping nodes
om	Number of communities to which overlapping nodes belong
μ	Mixed parameter

TABLE 4. Parameter settings of the artificial networks.

Parameter	Description
T_1	N=1000, k=20, k_{max}=50, C_{min}=10, C_{max}=100, μ=0.1~0.6
T_2	N=1000, k=20, k_{max}=50, C_{min}=10, C_{max}=100, μ=0.3, om=2, on=10~100
T_3	N=2000~10000, k=20, k_{max}=50, C_{min}=10, C_{max}=50, μ=0.3, om=4

2) ARTIFICIAL DATASETS

The artificial networks are generated by the LFR [40] benchmark to verify the performance of the CDCLM algorithm. The parameters of the LFR artificial networks are shown in Table 3.

Three sets of artificial networks are used in the experiments. The parameter settings are shown in Table 4.

B. EXPERIMENTAL SCHEME AND EVALUATION METRICS

1) EXPERIMENTAL SCHEME

In the experiments, five classical comparison algorithms are selected, namely, CDTD algorithm [30], Attractor algorithm [27], MCL algorithm [41], CPM algorithm [42], and DEMON algorithm [43], to verify the performance of the CDCLM algorithm. We compare and analyze the experimental results of the algorithms on real-world networks and artificial networks. The parameter settings of the above algorithms are shown in Table 5.

2) EVALUATION METRICS

Some assessments of a partitioned community structure are researched [34]. In the experiments, the overlapping modularity EQ [34] is selected as the evaluation metric. The closer the

TABLE 5. Settings of the algorithms' parameters.

Algorithm	Parameter
CDCLM	δ=0.5~0.9, λ=0.6
CDTD	δ=0.5~0.9, λ=0.6
Attractor	λ=0.6
MCL	*expand_factor* = 2, *inflate_factor* = 2, *max_loop* = 20, *mult_factor* = 0.5
CPM	k=3~8
DEMON	e=0.1~0.5

value of EQ is to 1, the higher the quality of the communities discovered by the algorithm. The closer the value of EQ is to 0, the worse the quality of the communities discovered. EQ is calculated according to Equation (11).

To compare the accuracy of the above algorithms on the networks whose true communities are given, the normalized mutual information NMI [44] is used. The closer the value of NMI is to 1, the higher the accuracy of the algorithm. The closer the value of NMI is to 0, the lower the accuracy of the algorithm. The equation of NMI is as follows:

$$NMI = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} N_{ij}\log(\frac{N_{ij}N}{N_{i\cdot}N_{\cdot j}})}{\sum_{i=1}^{C_A} N_{i\cdot}\log(\frac{N_{i\cdot}}{N}) + \sum_{j=1}^{C_B} N_{\cdot j}\log(\frac{N_{\cdot j}}{N})}, \quad (17)$$

where N is the confusion matrix; N_{ij} is the element of the i-th row and the j-th column in N, and its value represents the number of common nodes of community i and community j, respectively; $N_{i\cdot}$ is the sum of the elements of the ith row of the matrix; $N_{\cdot j}$ is the sum of the elements of the jth column of the matrix; C_A is the number of real communities; and C_B is the number of communities found by the algorithm.

C. EXPERIMENTS OF ALGORITHM'S PARAMETERS

Different values of parameter δ of the CDCLM algorithm will affect the accuracy of the algorithm. Thus, experiments are conducted to determine the proper value of the parameter. Network T_1 is used in the experiments. The experimental results are shown in Figure 2(a) illustrates that the accuracy of the CDCLM algorithm increases rapidly and stabilizes soon as the value of δ increases. The algorithm gets high precision when $\delta = 0.2 - 0.6$. As shown in Figure 2(b), as the value of δ increases, the accuracy of the algorithm increases gradually. The accuracy of the algorithm is high when $\delta = 0.6 - 0.94$. In summary, if the parameter δ is set to 0.6, then the precision of the CDCLM algorithm is always at a high level. Hence, the value of parameter δ is set to 0.6 in subsequent experiments.

D. EXPERIMENTS OF ALGORITHM PRECISION

1) EXPERIMENTAL RESULT ON THE REAL-WORLD DATASETS

Table 6 shows the experimental results of modularity of the CDCLM, CDTD, Attractor, MCL, CPM, and DEMON

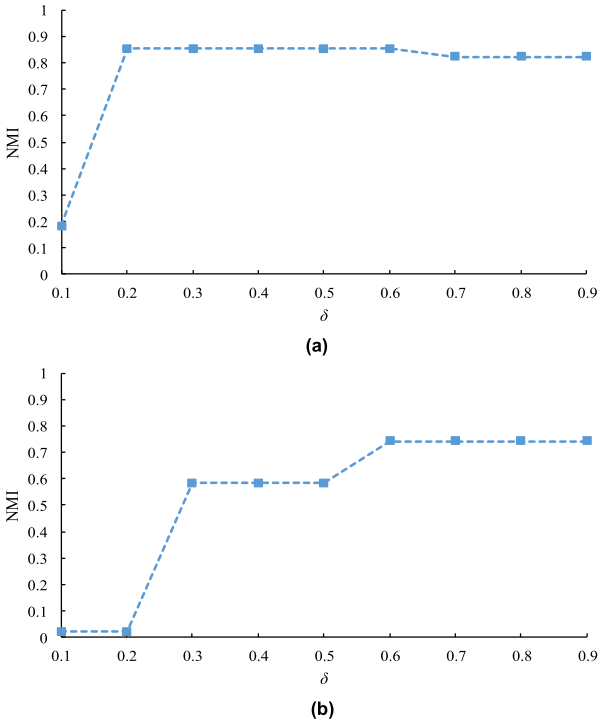


FIGURE 2. Experimental results on parameter δ. (a) $\mu = 0.2$. (b) $\mu = 0.4$.

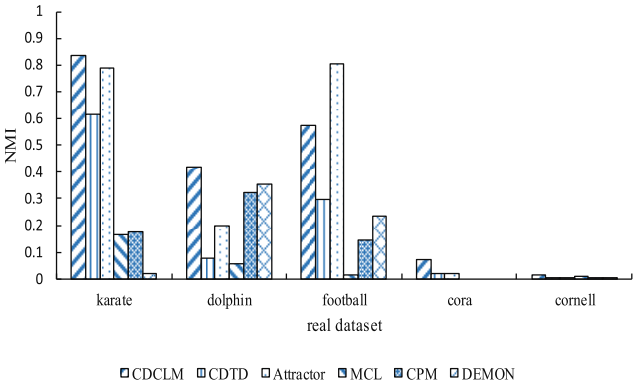


FIGURE 3. Experimental results of NMI on real-world dataset.

algorithms on the real-world datasets. As shown in the table, the CDCLM algorithm is better than the comparison algorithms on most datasets. The EQ values of the CDCLM algorithm are not as good as the CDTD and Attractor algorithms on Karate and Dolphin. However, as shown in Figure 3, the NMI values of the CDCLM algorithm on these two datasets are superior to those of CDTD and Attractor because the NMI metric is more objective than the EQ metric. On the basis of the results, our algorithm can find more precise communities than the other two algorithms.

Figure 3 shows the experimental results of NMI of the CDCLM, CDTD, Attractor, MCL, CPM, and DEMON algorithms on the four real-world datasets. As shown in the figure, the CDCLM algorithm performs the best on

TABLE 6. Overlapping modularity experimental results on real-world sets.

Real-world dataset	CDCLM	CDTD	Attractor	MCL	CPM	DEMON
Karate	**0.4216***	**0.4451**	0.4348	0.1898	0.2294	0.2456
Dolphin	**0.3658***	0.0569	**0.4452**	0.0110	0.3925	0.1259
Football	0.5308	0.3162	**0.5753**	0.0165	0.1116	0.1848
Texas	**0.5246**	0.5194	0.5135	0.0460	0.2038	0.0563
Cornell	**0.5061***	0.4923	**0.6246**	0.0038	0.2599	0.2044
Cora	**0.6953**	0.6799	0.6662	0.0073	0.3696	0.2668
Power	**0.8311**	0.8084	0.2071	0.2261	0.1566	0.0877
CA-GrQc	0.6716	**0.7057**	0.6845	0.4356	0.4744	0.3520

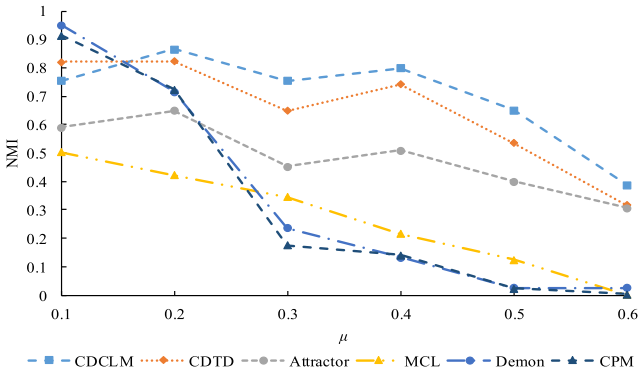


FIGURE 4. Experimental results of NMI on T_1.

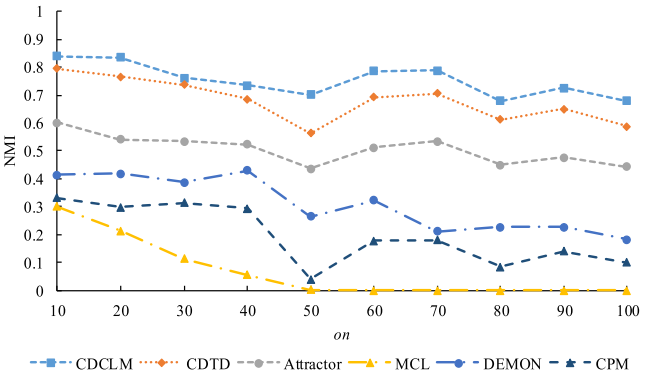


FIGURE 5. Experimental results of NMI on T_2.

most datasets. This result is largely because it uses the mechanism of dynamic distance to update the distance between nodes and finds overlapping community structure on the basis of incremental modularity maximum. Therefore, the CDCLM algorithm can find overlapping communities with high precision.

2) EXPERIMENTAL RESULT ON THE ARTIFICIAL DATASETS

a: EXPERIMENTS WITH DIFFERENT VALUES OF μ

Figure 4 shows the experimental results of the algorithms on artificial network T1. As shown in the figure, with the increase of the value of μ, the NMI values of all the algorithms gradually decrease. When the value of μ increases to a certain value, the NMI value drops dramatically. The boundaries of the communities are getting blurred as the value of μ increases. As a result, most algorithms experience difficulty when identifying communities accurately.

In Figure 4, except for $\mu = 0.1$, the NMI values of the CDCLM algorithm are higher than those of other algorithms. The CDTD and Attractor algorithms are in the second place. As the value of μ increases, the results of CDCLM become stable, and the fluctuation is small. This result is because the mechanism based on the dynamic distance considers the influence of three types of neighbor nodes when handling the influence of community nodes. Therefore, the robustness of the algorithm is improved.

b: EXPERIMENTS WITH DIFFERENT VALUES OF ON

Figure 5 shows the experimental results of the algorithms on artificial network T_2. As shown in the figure, as the value of *on* increases, the NMI value of each algorithm decreases slightly or remains unchanged.

In Figure 5, the CDCLM algorithm always obtains the best results no matter how the value varies. The results of CDTD and Attractor decrease slightly. This result is because the CDCLM algorithm uses incremental overlapping modularity and performs community optimization. As such, it can better discover the overlapping structures in the network than the other algorithms.

3) SCALABILITY EXPERIMENTS

In the scalability experiments, five classical algorithms are selected for comparison. Let the number of nodes of the original network be n, the number of edges be m, the average number of exclusive neighbors of two linked nodes be k, the number of time steps be T, and the maximum degree be K. The time complexity of CDCLM, CDTD, Attractor, MCL, DEMON and CPM is $O(n\log n + m)$, $O(n\log n + m)$, $O(m + km + Tm)$, $O(n^3)$, $O(nK^{3-\alpha})$ and $O(\alpha n^{\beta \ln(n)})$, separately, where α, β are constants. In most cases, T satisfies $3 \leq T \leq 50$. The time complexity of CDCLM and CDTD is generally not very high.

Figure 6 shows the experimental results of the algorithms on artificial network T_3. It reflects the time cost of each

TABLE 7. Scalability experimental results on real-world sets.

Real-world dataset (seconds)	CDCLM	CDTD	Attractor	MCL	CPM	DEMON
Football	0.6070	0.8144	5.2324	0.7391	0.0095	0.4045
Cora	0.4108	0.4268	143.0263	117.4111	0.1211	3.5432
Power	0.6808	0.7152	32.5664	635.0319	0.0993	3.0099
CA-GrQc	1.2478	1.3488	530.9767	724.8932	0.2492	18.0997
Astro-ph	43.0532	43.5142	50165.5486	9413.2292	7.2263	233.7444
CA-CondMat	20.4324	21.6544	24651.0068	4647.2533	2.3496	113.5391

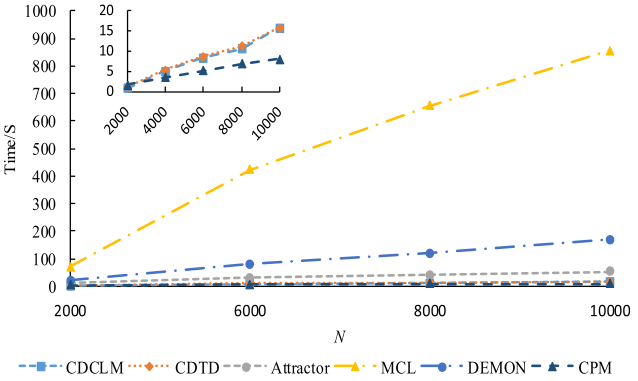


FIGURE 6. Experimental results of running time on T_3.

algorithm when the size of the data set increases. The figure in the upper-left corner of Figure 6 is the time comparison diagram after removing the Attractor, MCL, and DEMON algorithms. As shown in Figure 6, as the value of N increases, the time cost of each algorithm rises as well. The CDCLM algorithm also performs well. Its time cost increases linearly with the increase of the size of the datasets. This result is consistent with the analysis of time complexity of the algorithm in part D of Section IV. The CDCLM algorithm adopts the strategy based on triangle coarsening. In this manner, it can greatly reduce the network size while maintaining the community information as much as possible. Therefore, the running time of the algorithm can reduce greatly. The CDTD, Attractor, CPM, and DEMON algorithms perform relatively well. Although the time complexity of CPM is not the lowest, it is very suitable for networks with many complete subgraphs, that is, networks with dense edges. By contrast, the MCL algorithm performs poorly due to its high time complexity.

Table 7 shows the experimental results of the algorithms on real-world datasets. The results are consistent with those on the artificial networks. As shown in Table 7, CDCLM, CDTD, CPM perform the best, followed by DEMON, Attractor and MCL.

VI. CONCLUSION

We propose the CDCLM algorithm in this paper. First, the triangle-based coarsening strategy is adopted to reduce the network scale. Second, the initial community detection is performed on the coarsened network, and the mechanism of Attractor with dynamic distance is adopted. Third, the initial non-overlapping community structures are obtained by the reverse coarsening. Finally, the overlapping structure in the network is detected by the method based on local incremental overlapping modularity. Then, the new intimacy calculation strategy is used to optimize the community structure. Experimental results show that the CDCLM algorithm can find overlapping communities with high precision while maintaining near linear time complexity. In future studies, we will improve the performance of the CDCLM algorithm based on the incremental analysis strategies and apply it to detect communities in the dynamic social networks. In addition, the MapReduce model will also be tried to parallelize the CDCLM algorithm for discovering communities in large networks.

REFERENCES

[1] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, Feb. 2004, Art. no. 026113.

[2] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, Sep. 2007, Art. no. 036106.

[3] S. Gregory, "Finding overlapping communities in networks by label propagation," *New J. Phys.*, vol. 12, no. 10, Oct. 2010, Art. no. 103018.

[4] J. Xie and B. K. Szymanski, "Community detection using a neighborhood strength driven label propagation algorithm," in *Proc. IEEE Netw. Sci. Workshop*, West Point, NY, USA, Jun. 2011, pp. 188–195.

[5] C.-L. Zhang, Y.-L. Wang, Y.-J. Wu, B.-Y. Su, and X.-D. Wang, "Multi-label propagation algorithm for overlapping community discovery based on information entropy and local correlation," *J. Chin. Comput. Syst.*, vol. 37, no. 8, pp. 1645–1650, Aug. 2016.

[6] K. Deng, W. P. Li, F. H. Yu, and J. P. Zhang, "Overlapping community detection in complex networks based on multi kernel label propagation," *J. Commun.*, vol. 38, no. 2, pp. 53–66, Feb. 2017.

[7] H.-J. Li, Z. Bu, Z. Wang, J. Cao, and Y. Shi, "Enhance the performance of network computation by a tunable weighting strategy," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 3, pp. 214–223, Jun. 2018.

[8] M. E. J. Newman and E. A. Leicht, "Mixture models and exploratory analysis in networks," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 23, pp. 9564–9569, Jun. 2007.

[9] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *J. Mach. Learn. Res.*, vol. 9, pp. 1981–2014, Sep. 2008.

[10] E. P. Xing, W. Fu, and L. Song, "A state-space mixed membership blockmodel for dynamic network tomography," *Ann. Appl. Statist.*, vol. 4, no. 2, pp. 535–566, 2010.

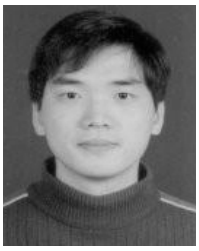
[11] Y. Xin, Z.-Q. Xie, and J. Yang, "An adaptive random walk sampling method on dynamic community detection," *Expert Syst. Appl.*, vol. 58, pp. 10–19, Oct. 2016.
[12] J. B. Pereira-Leal, A. J. Enright, and C. A. Ouzounis, "Detection of functional modules from protein interaction networks," *Proteins Struct. Function Bioinf.*, vol. 54, no. 1, pp. 49–57, 2004.
[13] C. Shi, Y. Cai, D. Fu, Y. Dong, and B. Wu, "A link clustering based overlapping community detection algorithm," *Data Knowl. Eng.*, vol. 87, no. 9, pp. 394–404, Sep. 2013.
[14] M. Zhu, F. Meng, and Y. Zhou, "Density-based link clustering algorithm for overlapping community detection," *J. Comput. Res. Develop.*, vol. 50, no. 12, pp. 2520–2530, Dec. 2013.
[15] D. He, D. Jin, C. Baquero, and D. Y. Liu, "Link community detection using generative model and nonnegative matrix factorization," *PLoS ONE*, vol. 9, no. 1, Jan. 2014, Art. no. e86899.
[16] K. Guo, E. B. Chen, and W. Z. Guo, "Overlapping community detection based on edge density clustering," *Pattern Recognit. Artif. Intell.*, vol. 31, no. 8, pp. 693–703, Aug. 2018.
[17] J. Baumes, M. Goldberg, M. Krishnamoorthy, M. Magdon-Ismail, and N. Preston, "Finding communities by clustering a graph into overlapping subgraphs," in *Proc. IADIS AC*, Faro District, Portugal, 2005, pp. 97–104.
[18] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New J. Phys.*, vol. 11, no. 3, Mar. 2008, Art. no. 033015.
[19] R. Kanawati, "LICOD: Leaders identification for community detection in complex networks," in *Proc. SocialCom/PASSAT*, Boston, MA, USA, 2011, pp. 577–582.
[20] R. Kanawati, "Seed-centric approaches for community detection in complex networks," in *Social Computing and Social Media* (Lecture Notes in Computer Science). New York, NY, USA: Springer, 2014, pp. 197–208.
[21] Z. Yu, J. Chen, K. Quo, Y. Chen, and Q. Xu, "Overlapping community detection based on random walk and seeds extension," in *Proc. Chin. Conf. Comput. Supported Cooperat. Work Social Comput.*, Chongqing, China, 2017, pp. 18–24.
[22] Y. Su, B. Wang, and X. Zhang, "A seed-expanding method based on random walks for community detection in networks with ambiguous community structures," *Sci. Rep.*, vol. 7, Feb. 2017, Art. no. 41830.
[23] H.-J. Li, Z. Bu, A. Li, Z. Liu, and Y. Shi, "Fast and accurate mining the community structure: Integrating center locating and membership optimization," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2349–2362, Sep. 2016.
[24] W. Chen, Z. Liu, X. Sun, and Y. Wang, "A game-theoretic framework to identify overlapping communities in social networks," *Data Mining Knowl. Discovery*, vol. 21, no. 2, pp. 224–240, 2010.
[25] Z. Bu, H.-J. Li, J. Cao, Z. Wang, and G. Gao, "Dynamic cluster formation game for attributed graph clustering," *IEEE Trans. Cybern.*, vol. 49, no. 1, pp. 328–341, Jan. 2017.
[26] Z. Bu, H.-J. Li, C. Zhang, J. Cao, A. Li, and Y. Shi, "Graph K-means based on leader identification, dynamic game and opinion dynamics," *IEEE Trans. Knowl. Data Eng.*, to be published.
[27] J. Shao, Z. Han, Q. Yang, and T. Zhou, "Community detection based on distance dynamics," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Sydney, NSW, Australia, 2015, pp. 1075–1084.
[28] T. Meng, L. J. Cai, T. He, L. Chen, and Z. Y. Deng, "An improved community detection algorithm based on the distance dynamics," in *Proc. Int. Conf. Intell. Netw. Collaborative Syst.*, Ostrava, Czech Republic, 2016, pp. 135–142.
[29] L. Chen, J. Zhang, and L.-J. Cai, "Overlapping community detection based on link graph using distance dynamics," *Int. J. Modern Phys. B*, vol. 32, no. 03, 2018, Art. no. 1850015.
[30] B. Xiang, K. Guo, Z. Liu, and Q. Liao, "An overlapping community detection algorithm based on triangle coarsening and dynamic distance," in *Proc. CCF Conf. Comput. Supported Cooperat. Work Social Comput.*, Guilin, China, 2018, pp. 285–300.
[31] J. M. Kumpula, M. Kivelä, K. Kaski, and J. Saramäki, "Sequential algorithm for fast clique percolation," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, no. 2, Aug. 2008, Art. no. 026109.
[32] H. Shen, X. Cheng, K. Cai, and M.-B. Hu, "Detect overlapping and hierarchical community structure in networks," *Phys. A, Stat. Mech. Appl.*, vol. 388, no. 8, pp. 1706–1712, Apr. 2009.
[33] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropol. Res.*, vol. 33, no. 4, pp. 452–473, Apr. 1977.
[34] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations," *Behavioral Ecol. Sociobiol.*, vol. 54, no. 4, pp. 396–405 Sep. 2003.
[35] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. United States Amer.*, vol. 99, no. 12, pp. 7821–7826, Apr. 2002.
[36] Q. Lu and L. Getoor, "Link-based classification," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, Washington, DC, USA, 2003, pp. 496–503.
[37] T. Yang, R. Jin, Y. Chi, and S. Zhu, "Combining link and content for community detection: A discriminative approach," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Paris, France, 2009, pp. 927–936.
[38] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.
[39] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Trans. Knowl. Discovery Data*, vol. 1, no. 2, Mar. 2007, Art. no. 2.
[40] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, no. 2, Oct. 2008, Art. no. 046110.
[41] S. Dongen, "A cluster algorithm for graphs," Centrum Wiskunde Inform., Amsterdam, The Netherlands, Tech. Rep. INS-R0010, 2000, pp. 1–42.
[42] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, Jun. 2005.
[43] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi, "Demon: A local-first discovery method for overlapping communities," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Beijing, China, 2012, pp. 615–623.
[44] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *J. Stat. Mech., Theory Exp.*, vol. 2005, no. 9, Sep. 2005, Art. no. P09008.

ZHANGHUI LIU is currently an Associate Professor with the College of Mathematics and Computer Science, Fuzhou University. He is also a member of the China Computer Federation (CCF) and the Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing. His research interests include big data technology and intelligence computation.

BINGJIE XIANG received the B.S. degree in network engineering from the Collage of Computer Engineering, Jimei University, Xiamen, China, in 2017. She is currently pursuing the M.S. degree with the School of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. Her research interest includes community detection.

WENZHONG GUO received the B.S. and M.S. degrees in computer science and the Ph.D. degree in communication and information system from Fuzhou University, Fuzhou, China, in 2000, 2003, and 2010, respectively, where he is currently a Full Professor with the College of Mathematics and Computer Science. He currently leads the Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing. His research interests include intelligent information processing, sensor networks, network computing, and network performance evaluation. He is also a member of ACM and a Senior Member of the China Computer Federation (CCF).

YUZHONG CHEN received the B.S. degree in computer engineering and the Ph.D. degree in information and communication engineering from the University of Science and Technology of China, Hefei, China, in 2000 and 2005, respectively. He is currently a Full Professor with the College of Mathematics and Computer Science, Fuzhou University. He is also a member of the CCF Young Computer Scientists & Engineers Forum and the Deputy Director of the Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing. His research interests include network information security, data mining, and social network analysis.



KUN GUO is currently an Associate Professor with the College of Mathematics and Computer Science, Fuzhou University. He is also a member of the China Computer Federation (CCF) and the Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing. His research interests include data mining, grey system theory, and distributed parallel computation.

JIANNING ZHENG received the B.S. degree from the Collage of Mathematics and Computer Science, Fuzhou University, Fuzhou, China, in 2003. He is currently the Vice-Chief Engineer with the Power Science and Technology Corporation State Grid Information and Telecommunication Group, Fuzhou. His research interest includes comprehensive energy services.